(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 767 585 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
09.04.1997 Bulletin 1997/15

(51) Int. Cl.^{6}: H04N 7/173

(21) Application number: 96305899.5

(22) Date of filing: 12.08.1996

(84) Designated Contracting States:
DE FR GB

(30) Priority: 02.10.1995 US 537025

(71) Applicant: International Business Machines Corporation
Armonk, N.Y. 10504 (US)

(72) Inventors:
- Dan, Asit
West Harrison, New York 10604 (US)
- Kienzle, Martin G.
Somers, New York 10589 (US)
- Sitaram, Dinkar
Yorktown Heights, New York 10598 (US)
- Tetzlaff, William H.
Mount Kisco, New York 10549 (US)

(74) Representative: Bailey, Geoffrey Alan
IBM United Kingdom Limited
Intellectual Property Department
Hursley Park
Winchester Hampshire SO21 2JN (GB)

(54) Video-on-demand systems

(57) A system and method for use in a distributed video-on-demand system of a type wherein at least one node (102,104,106,108) provides blocks (124) of video data to clients (111,112) and wherein at least some of the blocks (124) of video data are replicated on multiple nodes. Observed response characteristics for other nodes are recorded at least at a given one of the nodes which serves a client requesting a replicated block. The given one of the nodes also records response characteristics reported to it by the other nodes. The node from which to fetch the replicated data block is selected based on which nodes include a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

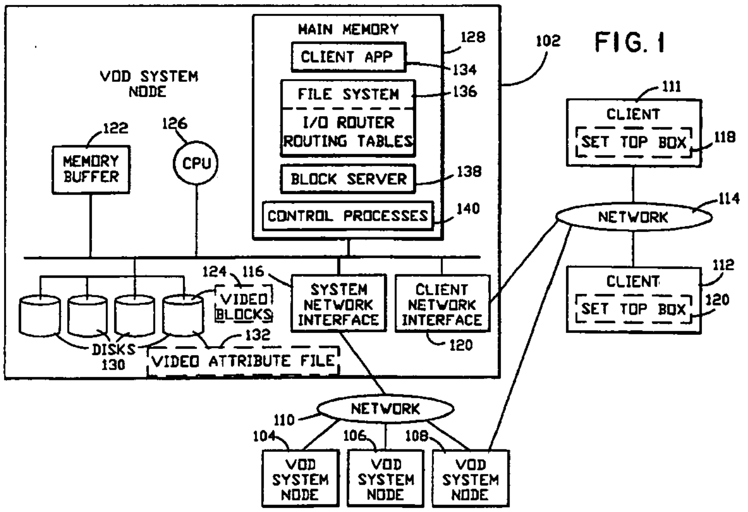


FIG. 1

Printed by Rank Xerox (UK) Business Services
2.14.1/3.4

Description

The present invention relates to a video on demand (VOD) system of the type wherein multiple clients are serviced by video streams delivered from a central video server.

A distributed real-time client server environment includes multiple clients and servers where each client's file system makes real-time I/O requests (i.e., requests with deadlines) to the servers responsible for retrieving the requested data. An example of such a system is a video server system consisting of multiple front-end client nodes with network adapters and back-end nodes or storage servers. In the presence of replication, the same data block may be present on multiple servers. At any random instant of time, the load (i.e., the number of requests waiting to be served or in service) on different servers is likely to be different. Additionally, different servers may have different performance characteristics (e.g., transfer speed, seek time, etc.).

A straightforward approach to server selection would be to randomly choose a server for an I/O operation. This approach, however, may require the servers to be run at a lower utilisation to avoid missing deadlines.

According to the first aspect of the present invention, there is provided in a distributed video server system of a type wherein at least one node provides video data to clients and wherein at least some of the data is replicated on multiple nodes, a method for selecting a node to serve the video data for provision to the clients, comprising the steps of:

recording observed response characteristics, the observed response characteristics being response information concerning nodes in the video server system as observed by a given node;

recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given node by at least one other node in the distributed video server system; and,

selecting a node to serve replicated video data based on which nodes include a copy of the replicated video data and based on at least one of the observed response characteristics and the reported response characteristics.

According to the second aspect of the present invention, there is provided a distributed video server system, comprising:

a plurality of nodes coupled by way of a first communication network;

a plurality of clients coupled to a given one of the nodes;

at least two of the nodes including a storage subsystem having replicated blocks of video data stored thereon;

means in the given one of the nodes, for recording observed response characteristics, the observed response characteristics being response information concerning nodes in the video server system as observed by the given one of the nodes;

means, in the given one of the nodes, for recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one of the other nodes; and

means, in the given one of the nodes, for selecting a node to serve the replicated data block based on which of the nodes includes a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

According to the third aspect of the present invention, there is provided in a video server system of a type wherein at least one node provides blocks of video data to clients and some of the blocks of video data are replicated on multiple storage devices, a method for selecting a storage device from which to retrieve the video data for provision to the clients, comprising the steps of:

recording observed response characteristics, the observed response characteristics being response information concerning performance of a plurality of the storage devices in the video server system as observed by the at least one node;

recording reported response characteristics, the reported response characteristics being response information concerning the plurality of the storage devices as reported to the at least one node in the video server system by computer processes managing the storage devices; and

selecting a storage device from which to retrieve the replicated data block based on which storage devices include a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

According to the fourth aspect of the present invention, there is provided a distributed computing system, comprising:

a plurality of nodes coupled by way of a first communication network;

a plurality of clients coupled to a given one of the nodes;

at least two of the nodes including a storage subsystem having replicated blocks of data stored thereon;

a first table instantiated in a memory in the given one of the nodes, dedicated to recording observed response characteristics, the observed response characteristics being response information concerning nodes in the system as observed by the given one of the nodes; and

a second table instantiated in the memory, dedicated to recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one other nodes.

According to the fifth aspect of the present invention, there is provided a method of controlling a distributed computing system of a type wherein a plurality of nodes are coupled by way of a first communication network and wherein a plurality of clients are coupled to a given one of the nodes, at least two of the nodes having the capability of performing an identical function, comprising the steps of:

recording observed response characteristics, the observed response characteristics being response information concerning nodes in the system as observed by the given one of the nodes;

recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one other nodes; and

selecting a node to perform the function based on which of the nodes can perform the function and based on at least one of the observed response characteristics and the reported response characteristics.

The present invention includes a system and method wherein clients use information about the observed server response time, and server performance measures to select a node to perform a given replicated function. The function can be, for example, provision of a data block available from more than one node. In a preferred embodiment, both the clients and the servers share server performance measures by piggybacking this information on normal messages passed between the clients and the servers as well as those passed between one server and another.

The invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram of a distributed video-on-demand system according to an embodiment of the present invention;

Figs. 2A-2C show the data structures used by the system of Fig. 1;

Fig. 3A and 3B show the procedure used by the client to select a server;

Fig. 4 shows the procedure used by the client to update the server performance measures;

Fig. 5 shows the procedure used by the client to update the observed server response time; and,

Fig. 6 shows the format of a message including piggy backed performance data.

The present system and method schedules real-time I/O operation so as to improve VOD system throughput while satisfying real-time service requirements. The approach takes into account the observed storage server response time (i.e the response time of the storage server that has the requested data), together with performance measures such as the expected storage server load and block placement among the storage servers, to select a block replica to be read. Each client has a corresponding file system (which can be shared with other clients). Whenever a response is received from a storage server, the file systems update a local copy of the observed storage server's response times and the time of day at which each response was observed.

The expected load information on different storage servers is maintained in the following manner. Both the file systems and the storage servers maintain an array of load counts on each storage server and the associated local timestamp at the storage server. The file systems and the storage servers need not have synchronised clocks. This information is piggybacked on every message between the file systems and the storage servers. On receiving this information each node (file system or storage server) updates its load count array to reflect the latest load information for a storage server based on the associated timestamp. Using the above mechanism a file system obtains load information even about those storage servers it has not accessed recently. It should be understood that each node may contain a file system, a storage server or both.

Fig. 1 is a block diagram of a distributed video-on-demand system according to an embodiment of the present invention. Computer systems (VOD System nodes) 102-108 are interconnected by way of a first (system) communication network (or switch) 110. The computer systems can be of the same or different types and can have different performance characteristics (from one another) depending on system type and configuration. One or more of the VOD nodes is also interconnected with clients 111, 112 by way of a second (client) communication network 114. Each VOD System Node includes a system network interface 116 which provides an electronic and communication interface to the system communication network 110. Nodes connected to the client communication network also include a client network interface 120 which provides an electronic and communication interface to the client communication network 114.

Each VOD System Node 102-108 includes a buffer memory 122 for the temporary storage of retrieved video blocks 124 and other messages and a processor (cpu) 126 which operates under control of various programs residing in a main memory 128. At least some of the nodes 102-108 include disks 130 which have movies stored thereon. Each of the movies is made up of a number of the video blocks 124. The video blocks 124 for a given video (e.g. movie) can be distributed (e.g. striped) across the disks 130 of a disk array within a single system or across the disks of multiple VOD system nodes. In the latter case, the video blocks are communicated from a VOD System Node that has a requested block on its disks to the VOD System Node serving a requesting client by way of the system communication network 110. The requested block is then sent to the requesting client by way of the client communication network 114. Attributes (size, space, play history, block mapping, etc.) of each video are stored in an attribute file 132 which is also stored on one or more of the disks 130. Similar to the video-blocks, the attributes can also be communicated between VOD System Nodes by way of the system communication network 110.

The programs in the main memory will now be described. It should be understood that each of the programs causes the VOD System Node to perform a "process" under the program's direction. At least one of the computer systems has one or more client applications 134 which communicate with the client's 111, 112 by way of the client communication network 114. The client applications also start, stop, pause and resume video playback upon client request. Similarly, at least one of the computer systems includes one or more file systems 136 which manage block placement and retrieval of the video blocks on the disks. The file systems include an I/O Router 136a which is responsible for selecting the storage server from which to retrieve the requested block.

Systems with disks also include a block server program 138 which is responsible for the physical retrieval of blocks from the disks and for providing the blocks to a requesting filesystem (which could be on any of the VOD System Nodes, including those without local disks). The block server 138 is also referred to herein as a "storage server" or simply a "server".

The set of programs stored in the main memory 128 also include control process modules 140 that, on various nodes, reserve a channel (i.e., resources) and set up viewing sessions before the start of video playback. Those of skill in the art will recognise that a number of other conventional software controlled processes, not described in detail here, are also involved in the control and support of the video server functions.

The VOD System Nodes 102-108 can be embodied using any processor of sufficient performance for the number of video streams to be supported. For example, a small capacity video server could be embodied using a RISC System/6000 TM system while a larger capacity server could be embodied using an ES/9000 TM system (both available form International Business Machines Corporation of Armonk, New York). The disks 130 can be embodied as any conventional disk subsystem or disk array. The communication networks 110, 114 can be, for example, fibre optic networks, conventional bidirectional cable networks or any other network of sufficiently high bandwidth to handle the real-time requirements of video data delivery in a VOD System. The clients can be embodied as set-top boxes 118, 120, or workstations or a combination of both.

Fig. 2A shows the Server Load Table 200. The Server Load Table is a data structure maintained by both the block server 138 and the file systems 136 on each node for tracking load information about the block servers. Each VOD System Node maintains its own copy of the Server Load Table. Each copy is updated to reflect the latest loads by way of piggyback information sent along (from one node to another) with video data blocks. Specifically, a copy of the table is piggybacked by the file systems onto each request and by the block servers onto each response.

Each entry in the Server Load Table 200 tracks (stores) the observed response time and load of each block server 138 in the VOD System Node. Each row contains the serverId 210, the observed delay 230 of the last response from that server, the delay timestamp 220 containing the time at which the delay 230 was observed, and the server load 250 as reported by the server (e.g. server utilisation). The load timestamp 240 is the server generated time stamp of the time at which the load 250 was reported. The load can be, for example, defined as the queue length (number of outstanding requests) at the server or as the server utilisation measured as the number of requests served per unit time.

The file system also maintains routing tables 136b. The Routing Tables include a Video Block Table 260 (shown in more detail in Fig. 2B) and a Request Table 285 (shown in more detail in Fig. 2C). The Video Block Table 260 contains a row for each video block. Each row contains the file identifier (fileId) 265 for the video block, the video block number (blockNo) 270, the number of replicas 275 of the video block and a list (Serv. Id) 280 of the storage servers (and therefore the VOD System Nodes) that have the video block. In the example shown, file F1 block b81 has 1 replica on server S6 whereas file F2 block b95 has three replicas on servers S3, S1, S10.

The Request Table 285 is used to update the Delay fields 230 of the Server Load Table 200. Each row of the Request Table corresponds to an outstanding request and contains an identifier for the request (requestId) 290. The Request Time field (requestTime) 295 is the time (according to the file system's clock) that the request was sent to the server and is set by the file system when the request is sent.

The steps used by the file system to select a server when it is necessary to request a new file block are shown in Figs. 3A-3B. The server table 200 contains the observed response time and the reported server load. Both types of information are weighted as follows by appropriate confidence factors when deciding which server to select. In step 302, the file system searches the Video Block Table 260 using the fileId 265 and blocKNos 270 of the requested block to find the number of replicas 275 (denoted by n) and the server ids 280

(S1,...,Sn) of the servers on which these blocks reside. In step 304, the file system computes a delay confidence factor CFi,d for each server Si based on the current time t and the delay timestamp 220 (Ti,d). This factor measures the level of confidence of the file system that the observed delay 230 is still current. The delay confidence factor is low if Ti,d is in the remote past. In the present embodiment, the file system computes $Cfi,d = 1/(e^x)$ where x is given by (t-Ti,d)/T and T is a pre-determined scaling factor. Similarly, in step 306 the file system computes a load confidence factor Cfi,u for each server based upon the load timestamp 240 as $Cfi,u = 1/(e^y)$ where y is given by (t-Ti,u)/T .

In step 308, the file system computes an overall delay confidence factor CFd as follows. It first computes Pd, the product of all the server delay confidence factors Cfi,d. Note that Pd will be high only if all the individual Cfi,d are high. Similarly, the file system computes Pu, the product of all the Cfi,u. Cfd is then computed as wd*Pd/(wd*Pd+wu*Pu) where wd and wu are weights that indicate the relative importance of the delay 240 and the load 250. If both are to be given equal importance, wd=wu=1 . In step 310 the load confidence factor CFu is computed as wu*Pu/(wd*Pd+wu*Pu) .

In step 312, the file system computes a delay badness factor Bi,d for each server Si given by di/(d1+...+dn) . Bi,d is high if the delay of server i is high relative to the other servers. In step 314, the file system computes a load badness factor Bi,u for each Si given by ui/(u1+...un) . In step 316, the file system computes the overall badness factor Bi for each server Si as Cfi,d*Bi,d+Cfi,u*Bi,u . In step 318, the file system selects the server with the lowest badness factor Bi and exits in step 320.

Fig. 4 shows the steps used by the file systems and servers to update the loads 250 in the Server Load Table 200. These steps are executed whenever the file system receives a message from the server or when the server receives a request from the file system. In step 410, the server (or file system) extracts the update Server Load Table 610 (U) that was piggybacked on the request (or response). The format of the piggybacked message 600 including the update Server Load Table 610 is shown in Fig. 6. The update Server Load Table 610 is organised in the same way and includes the same fields as the Server Load Table 200.

In subsequent steps, the server (or file system) scans the local copy of the Server Load Table 200 (S) and updates the appropriate load 250. In step 405, the index i is set to the index of the first row in the Server Load Table (S). In step 410, the load timestamp 240 from table U 610 and the load timestamp 240 from table S 200 are compared. If the load timestamp 240 from table U is more recent, the server load 250 in table S is set to the value of server load (Load) 250 in table U. In step 420, a check is made to see if there are more servers in table S. If so, the index i is set to the next row and step 410 is executed. If not, the update load procedure terminates in step 430.

Fig. 5 shows the steps used by the file systems to update the delays 230 in the server load table 200. This step is executed whenever a response is received from a server. In step 510, the file system locates the entries in the Request Table 285 corresponding to this request. In step 520, the response time for this request is computed as the difference between the current time and the request time 295. In step 530, the file system locates the row corresponding to this server in the Server Load Table 200. The delay field 230 is set to the response time computed in step 520 and the delay timestamp field 220 is set to the current time.

While the present embodiment is described in conjunction with load information being monitored at the storage server level, the principles of the present invention can also be readily applied at the disk level. Thus, observed and reported disk performance can be used, even in the context of a single node video server, to select a disk from which to obtain a video data block replica.

Claims

1. In a distributed video server system of a type wherein at least one node provides video data to clients and wherein at least some of the data is replicated on multiple nodes, a method for selecting a node to serve the video data for provision to the clients, comprising the steps of:

   recording observed response characteristics, the observed response characteristics being response information concerning nodes in the video server system as observed by a given node;
   recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given node by at least one other node in the distributed video server system; and
   selecting a node to serve replicated video data based on which nodes include a copy of the replicated video data and based on at least one of the observed response characteristics and the reported response characteristics.

2. A method according to Claim 1, wherein the reported response characteristics are sent to the given node as information piggybacked on other messages passed between the nodes.

3. A method according to Claim 1 or 2, wherein the observed characteristics include a measured delay in retrieving the video blocks.

4. A method according to Claim 1, 2 or 3, wherein the reported characteristics include a measure of performance of load on each node.

5. A method according to Claim 4, wherein the measure of performance is reported along with a time stamp indicative of when the measure was taken.

6. A method according to Claim 5, wherein the measure of performance is stored by the given node and wherein only the measure of performance having a most recent time stamp for each node is retained by the given node.

7. A method according to Claim 3, wherein the replicated block is retrieved from the node having the shortest measured delay.

8. A method according to any one of Claims 1 to 6, wherein the observed response characteristics and the reported response characteristics are each given a weight and wherein the selecting is also based on the weight.

9. A method according to Claim 8, wherein the selecting is based on the response characteristics having the highest total weight.

10. A method according to any one of the preceding Claims, comprising the further step of providing the replicated data block to at least one of the clients.

11. A distributed video server system, comprising:

    a plurality of nodes (102,104,106,108) coupled by way of a first communication network (110);
    a plurality of clients (111,112) coupled to a given one of the nodes (102,104,106,108);
    at least two of the nodes (102,104,106,108) including a storage subsystem (130) having replicated blocks (124) of video data stored thereon;
    means (136,138,200) in the given one of the nodes (102,104,106,108), for recording observed response characteristics, the observed response characteristics being response information concerning nodes in the video server system as observed by the given one of the nodes;
    means (136,138,200), in the given one of the nodes (102,104,106,108), for recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one of the other nodes; and
    means (136,138), in the given one of the nodes (102,104,106,108), for selecting a node to serve the replicated data block (124) based on which of the nodes includes a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

12. A system according to Claim 11, wherein each of the nodes having a replicated copy of the block (124) includes means for sending the reported response characteristics to the given node as information piggybacked on other internode messages.

13. A system according to Claim 11 or 12, wherein the observed characteristics include a measured delay (230) in retrieving the video blocks (124).

14. A system according to Claim 12, wherein the reported characteristics include a measure of performance of load (250) on each node.

15. A system according to Claim 14, wherein the means for sending include means for including a time stamp (240) indicative of when the measure of performance (250) was taken.

16. A system according to Claim 15, wherein the given node includes means (200) for storing the measure of performance (250) and for retaining the measure of performance (250) having a most recent time stamp (240) for each node.

17. A system according to Claim 13, wherein the means (136,138) for selecting retrieves the replicated block (124) from the node having the shortest measured delay (230).

18. A system according to any one of Claims 11 to 16, wherein the observed response characteristics and the reported response characteristics are each given a weight and wherein the means (136,138) for selecting selects the node (102,104,106,108) at least in part based on the weight.

19. A system according to Claim 18, wherein the means (136,138) for selecting, selects the node (102,104,106,108) having the highest total weight.

20. In a video server system of a type wherein at least one node provides blocks of video data to clients and some of the blocks of video data are replicated on multiple storage devices, a method for selecting a storage device from which to retrieve the video data for provision to the clients, comprising the steps of:

    recording observed response characteristics, the observed response characteristics being response information concerning performance of a plurality of the storage devices in the video server system as observed by the at least one node;
    recording reported response characteristics, the reported response characteristics being response information concerning the plurality of the storage devices as reported to the at

least one node in the video server system by computer processes managing the storage devices; and,

selecting a storage device from which to retrieve the replicated data block based on which storage devices include a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

21. A distributed computing system, comprising:

a plurality of nodes (102,104,106,108) coupled by way of a first communication network (110);
a plurality of clients (111,112) coupled to a given one of the nodes (102,104,106,108);
at least two of the nodes (102,104,106,108) including a storage subsystem (130) having replicated blocks (124) of data stored thereon;
a first table (200) instantiated in a memory in the given one of the nodes (102,104,106,108), dedicated to recording observed response characteristics, the observed response characteristics being response information concerning nodes in the system as observed by the given one of the nodes; and
a second table (200) instantiated in the memory, dedicated to recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one other nodes.

22. A system according to claim 21 further comprising:

means (136,138) , in the given one of the nodes (102,104,106,108), for selecting a node to serve the replicated data block (124) based on which of the nodes includes a copy of the replicated data block and based on at least one of the observed response characteristics and the reported response characteristics.

23. A method of controlling a distributed computing system of a type wherein a plurality of nodes are coupled by way of a first communication network and wherein a plurality of clients are coupled to a given one of the nodes, at least two of the nodes having the capability of performing an identical function, comprising the steps of:

recording observed response characteristics, the observed response characteristics being response information concerning nodes in the system as observed by the given one of the nodes;
recording reported response characteristics, the reported response characteristics being response information concerning other nodes as reported to the given one of the nodes by at least one other nodes; and,
selecting a node to perform the function based on which of the nodes can perform the function and based on at least one of the observed response characteristics and the reported response characteristics.

FIG. 1

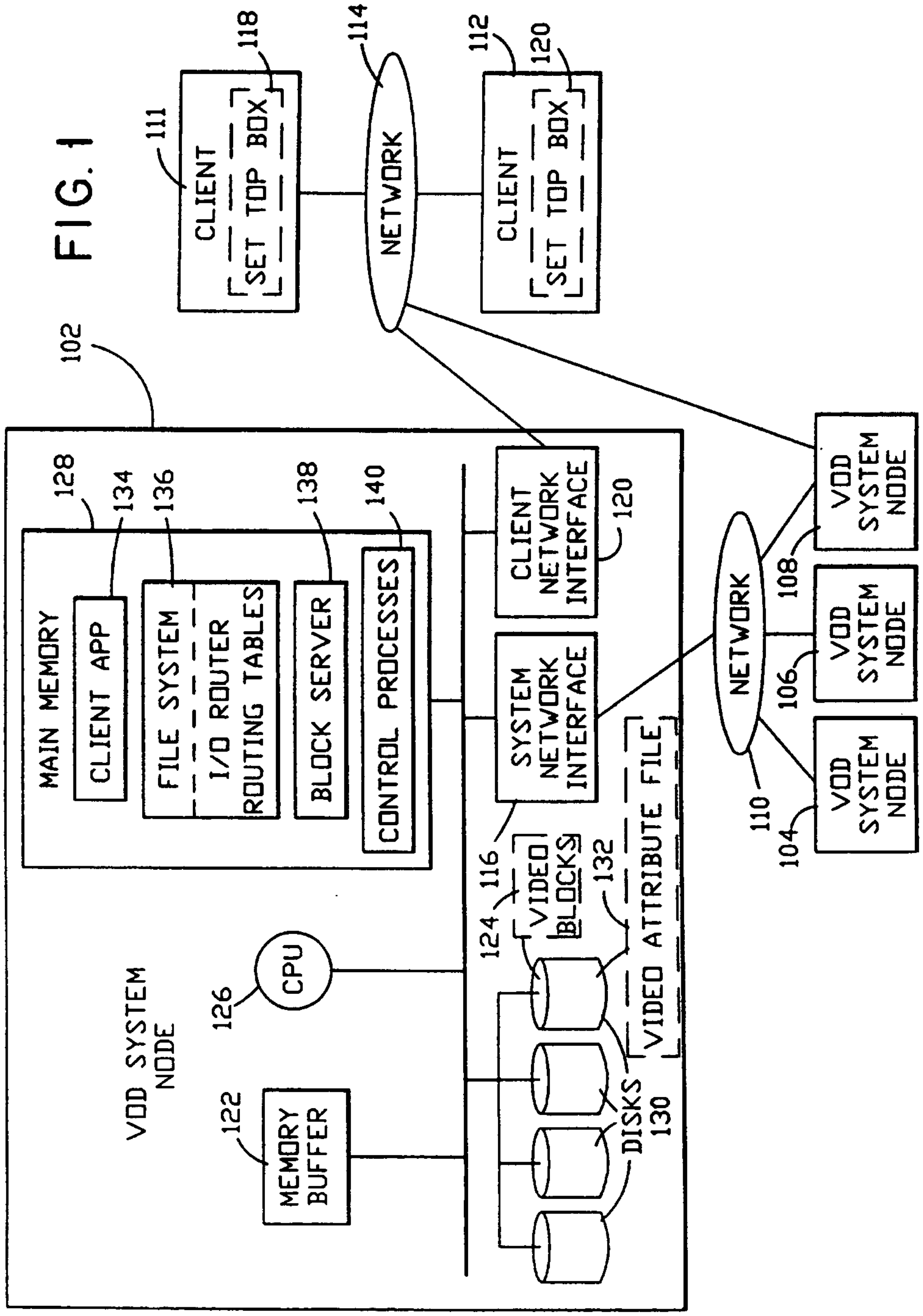
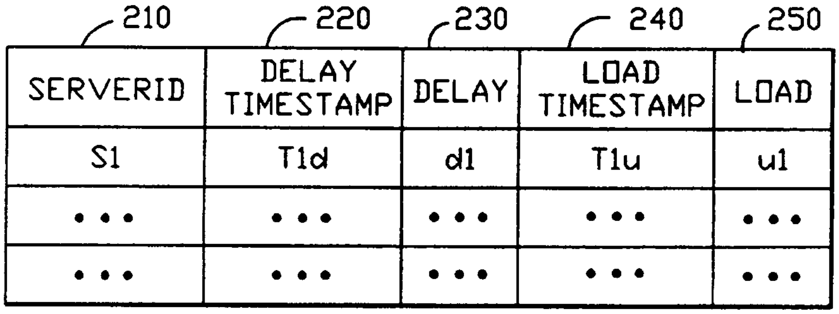

102
VOD SYSTEM NODE
128
MAIN MEMORY
134
CLIENT APP
136
FILE SYSTEM
I/O ROUTER
ROUTING TABLES
138
BLOCK SERVER
140
CONTROL PROCESSES
CLIENT NETWORK INTERFACE
120
SYSTEM NETWORK INTERFACE
116
122
MEMORY BUFFER
126
CPU
124
VIDEO BLOCKS
132
VIDEO ATTRIBUTE FILE
DISKS
130
111
CLIENT
SET TOP BOX
118
114
NETWORK
112
CLIENT
SET TOP BOX
120
NETWORK
110
VOD SYSTEM NODE
104
VOD SYSTEM NODE
106
VOD SYSTEM NODE
108

FIG. 2A

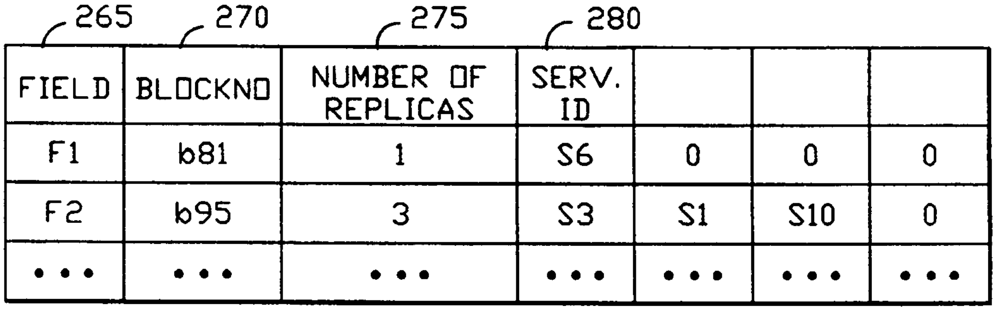
SERVER LOAD TABLE 200


210
220
230
240
250

SERVERID	DELAY TIMESTAMP	DELAY	LOAD TIMESTAMP	LOAD
S1	T1d	d1	T1u	u1
• • •	• • •	• • •	• • •	• • •
• • •	• • •	• • •	• • •	• • •

FIG. 2B

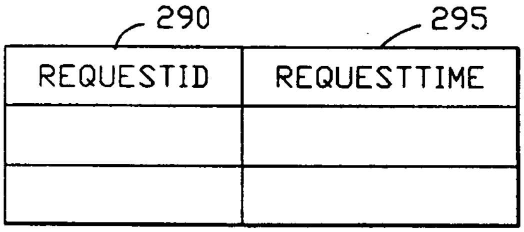
VIDEO BLOCK TABLE 260


265
270
275
280

FIELD	BLOCKNO	NUMBER OF REPLICAS	SERV. ID			
F1	b81	1	S6	0	0	0
F2	b95	3	S3	S1	S10	0
• • •	• • •	• • •	• • •	• • •	• • •	• • •

FIG. 2C

REQUEST TABLE 285



REQUESTID	REQUESTTIME

FIG. 3A

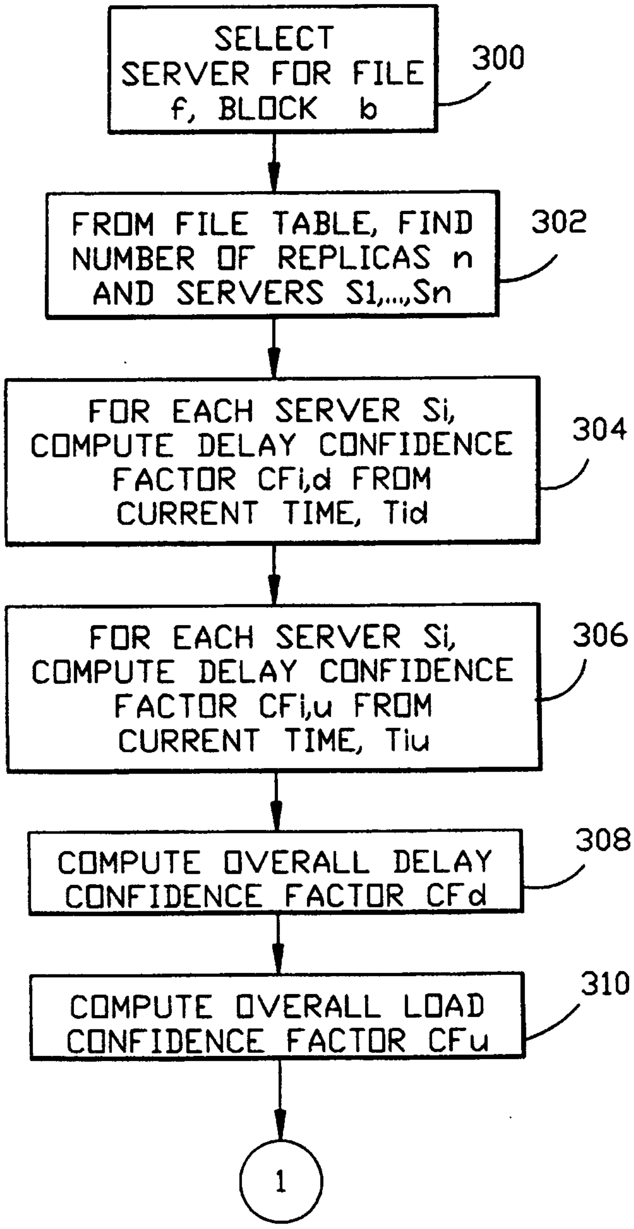

SELECT SERVER FOR FILE f, BLOCK b
300
FROM FILE TABLE, FIND NUMBER OF REPLICAS n AND SERVERS S1,...,Sn
302
FOR EACH SERVER Si, COMPUTE DELAY CONFIDENCE FACTOR CFi,d FROM CURRENT TIME, Tid
304
FOR EACH SERVER Si, COMPUTE DELAY CONFIDENCE FACTOR CFi,u FROM CURRENT TIME, Tiu
306
COMPUTE OVERALL DELAY CONFIDENCE FACTOR CFd
308
COMPUTE OVERALL LOAD CONFIDENCE FACTOR CFu
310
1

FIG. 3B

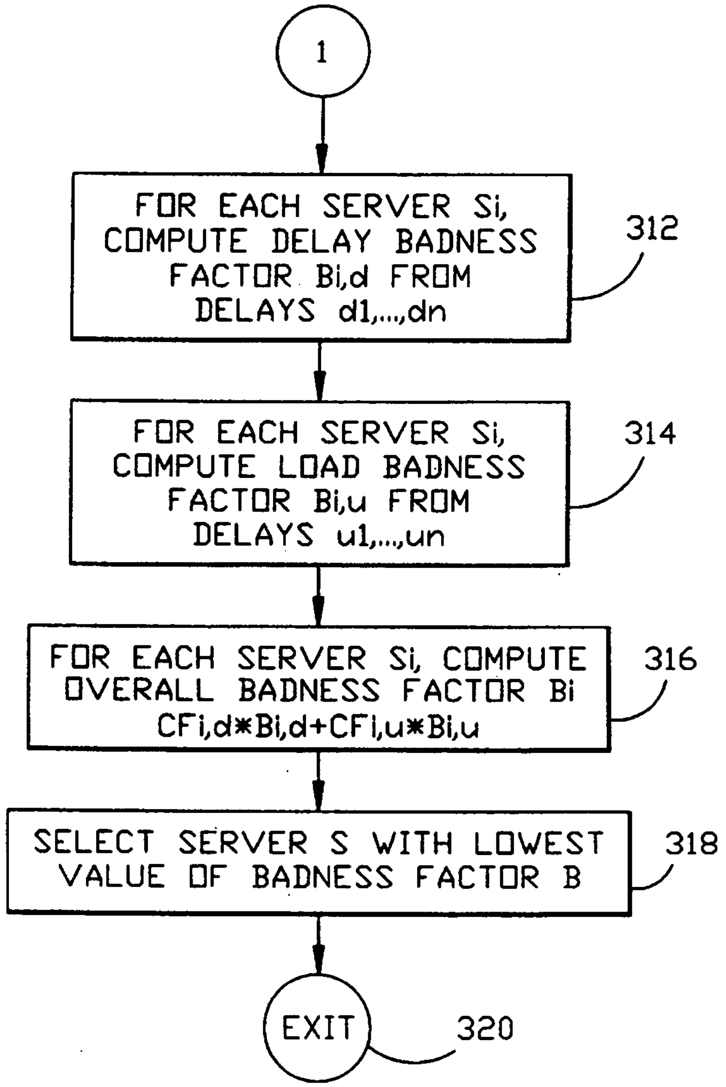

1
FOR EACH SERVER Si, COMPUTE DELAY BADNESS FACTOR Bi,d FROM DELAYS d1,...,dn
312
FOR EACH SERVER Si, COMPUTE LOAD BADNESS FACTOR Bi,u FROM DELAYS u1,...,un
314
FOR EACH SERVER Si, COMPUTE OVERALL BADNESS FACTOR Bi CFi,d*Bi,d+CFi,u*Bi,u
316
SELECT SERVER S WITH LOWEST VALUE OF BADNESS FACTOR B
318
EXIT
320

FIG. 4

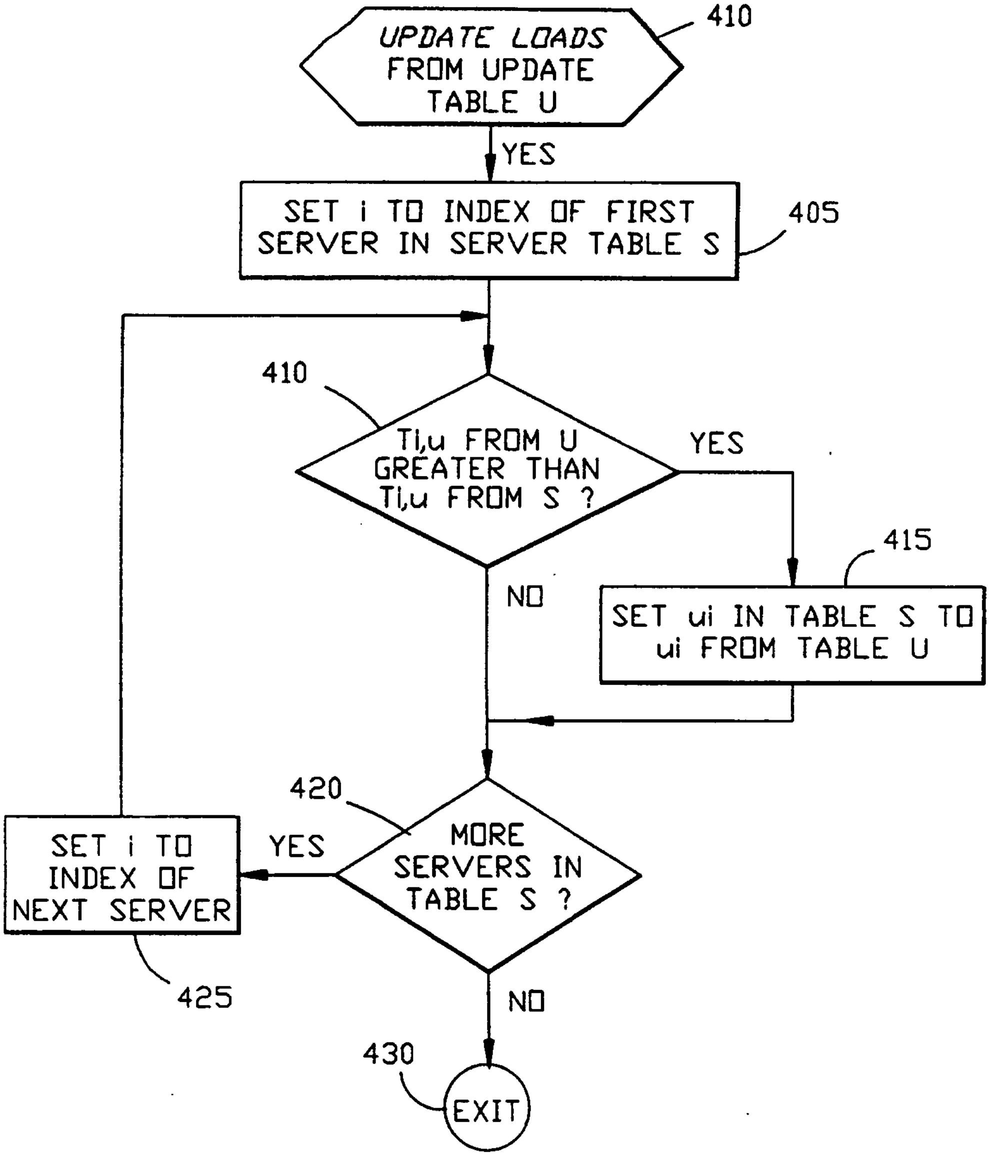

UPDATE LOADS FROM UPDATE TABLE U
410
YES
SET i TO INDEX OF FIRST SERVER IN SERVER TABLE S
405
410
Ti,u FROM U GREATER THAN Ti,u FROM S ?
YES
415
NO
SET ui IN TABLE S TO ui FROM TABLE U
420
MORE SERVERS IN TABLE S ?
YES
SET i TO INDEX OF NEXT SERVER
425
NO
430
EXIT

FIG. 5

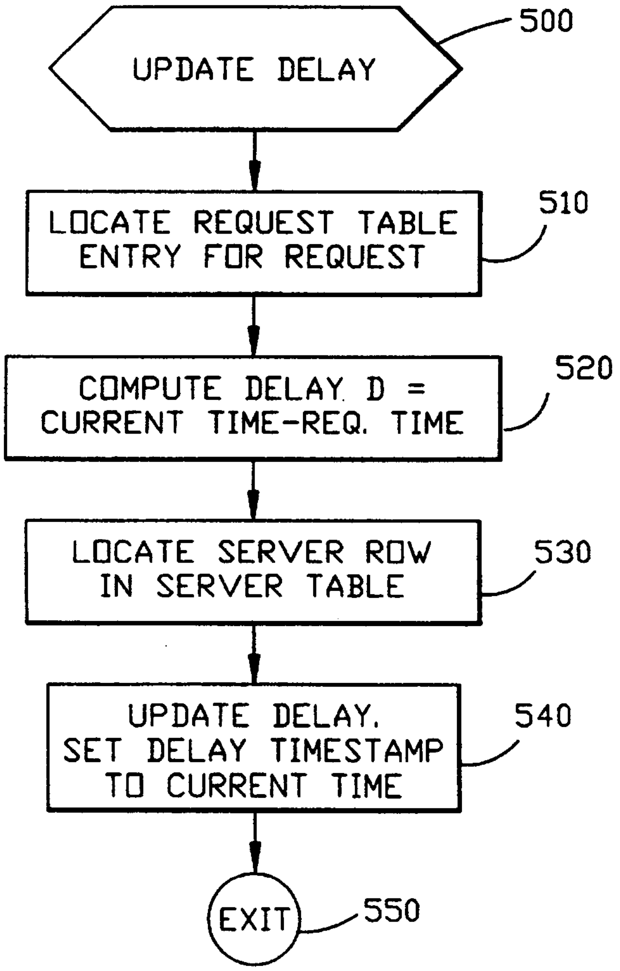
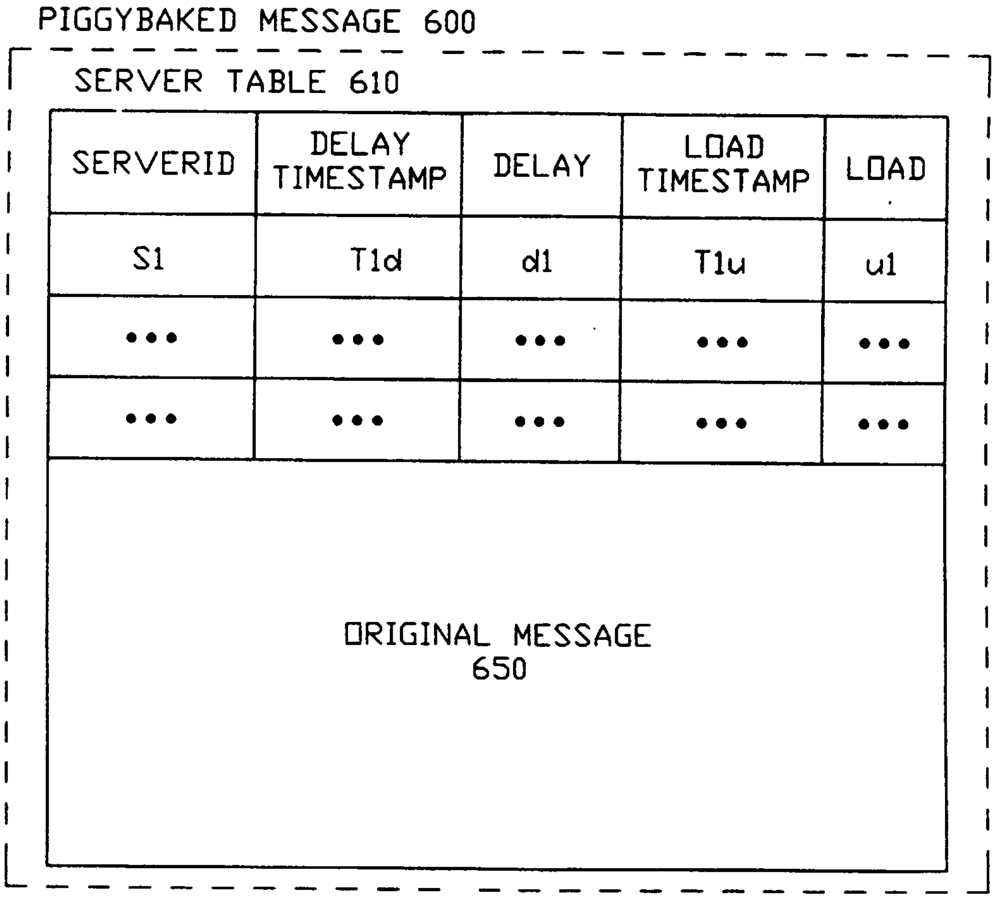

UPDATE DELAY
500
LOCATE REQUEST TABLE ENTRY FOR REQUEST
510
COMPUTE DELAY D = CURRENT TIME-REQ. TIME
520
LOCATE SERVER ROW IN SERVER TABLE
530
UPDATE DELAY. SET DELAY TIMESTAMP TO CURRENT TIME
540
EXIT
550

FIG. 6



SERVERID	DELAY TIMESTAMP	DELAY	LOAD TIMESTAMP	LOAD
S1	T1d	d1	T1u	u1
•••	•••	•••	•••	•••
•••	•••	•••	•••	•••